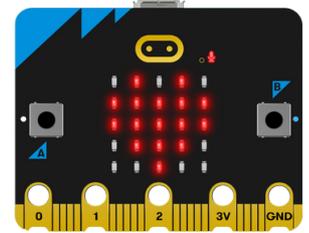


© Micro:bit Educational Foundation



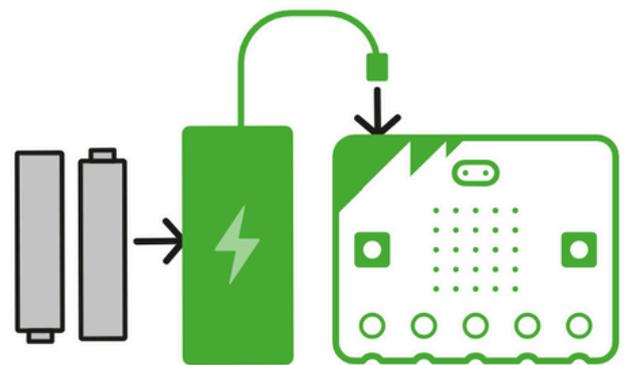
Der BBC micro:bit ist ein Computer im Taschenformat. Er verfügt über ein LED-Licht-Display, Tasten, Sensoren und viele Eingabe- und Ausgabe Funktionen, die programmiert werden können. Die neueste Version des micro:bit bietet zusätzlich die Möglichkeit, Töne zu erkennen und wiederzugeben.

Den micro:bit einschalten

Lege einfach 2 x AAA-Batterien in den Akkupack ein und schlieÙe ihn an deinen micro:bit an.

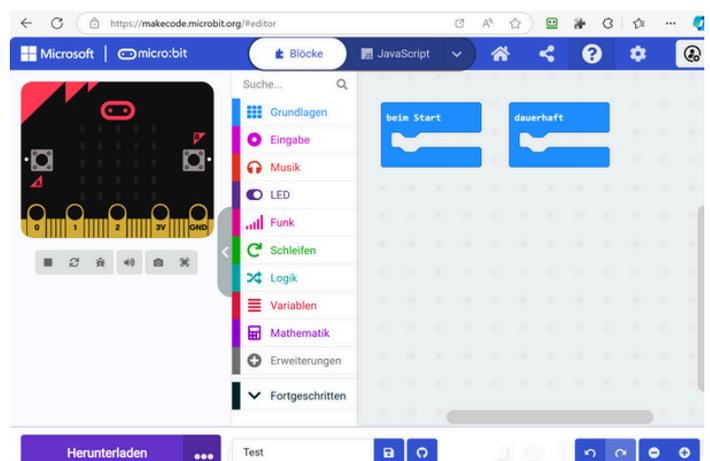
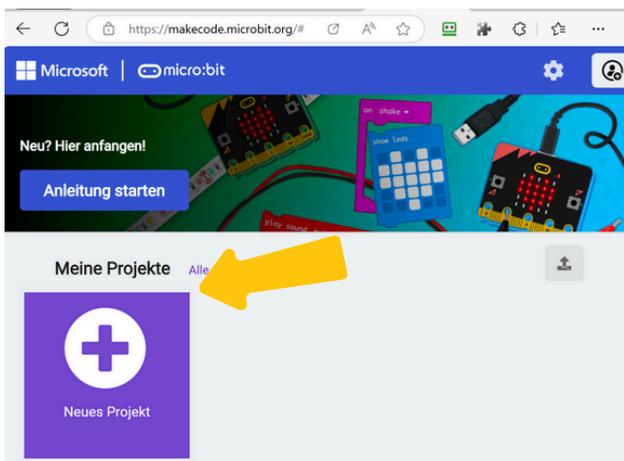
Dein micro:bit wird sich automatisch einschalten.

Hinweis: Du kannst deinen micro:bit auch mit dem USB-Kabel an einen Computer anschließen, ohne den Akku zu verwenden.



Programmieren: <https://makecode.microbit.org>

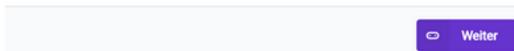
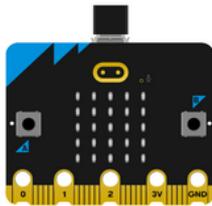
Der micro:bit kann mit der MakeCode Webseite auf jedem Computer programmiert werden. Der Code wird dann mit einem USB Kabel auf den micro:bit heruntergeladen. Der micro:bit wird über das USB Kabel mit Strom versorgt, kann aber auch mit dem Akkupack nach dem herunterladen ohne Kabel betrieben werden.





Wichtig: Bei jedem Herunterladen eines Programmes wird das alte Programm im micro:bit völlig ersetzt.

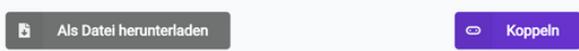
1. Verbinde deinen micro:bit mit deinem Computer



2. Koppel deinen micro:bit mit deinem Browser



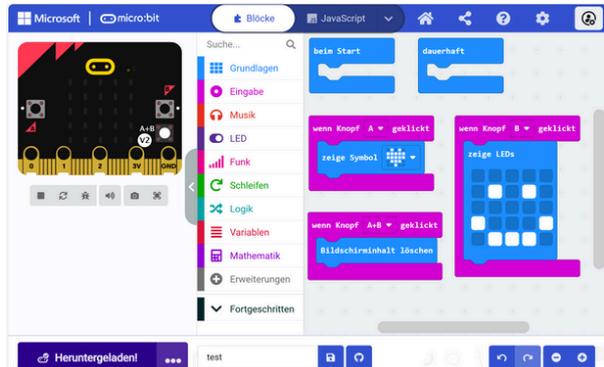
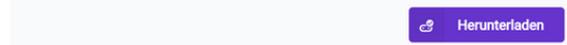
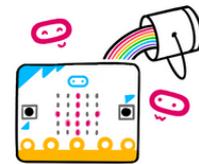
Klicke unten auf die Schaltfläche Koppeln.
Ein Fenster erscheint oben in deinem Browser.
Wähle das Gerät micro:bit und klicken auf Verbinden.



✓ Verbunden mit micro:bit



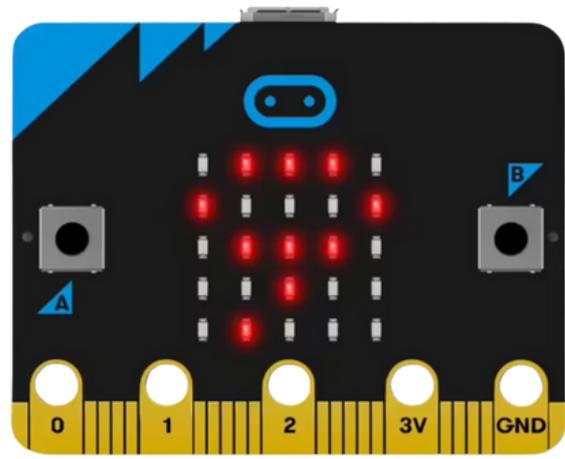
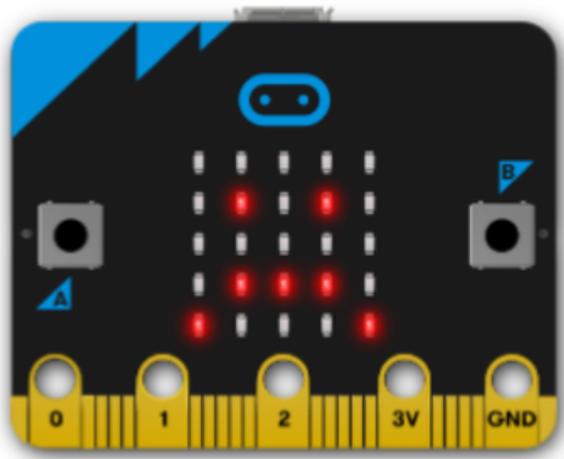
Dein Calliope mini ist verbunden! Durch Drücken von "Herunterladen" wird dein Code nun automatisch auf deinen Calliope mini kopiert.



Fehlerbehebung

Manchmal kommt es vor dass beim Herunterladen des Programmes auf den micro:bit ein Fehler passiert:

- **Verbindungsfehler:** Der Code wird nicht vollständig heruntergeladen. In MakeCode sieht man eine Meldung dass man das Programm speichern soll oder den micro:bit neu koppeln.
- **Programmfehler:** Der Code wurde heruntergeladen aber verursacht am micro:bit einen Fehler. Der micro:bit zeigt ein trauriges Gesicht und eine Nummer.



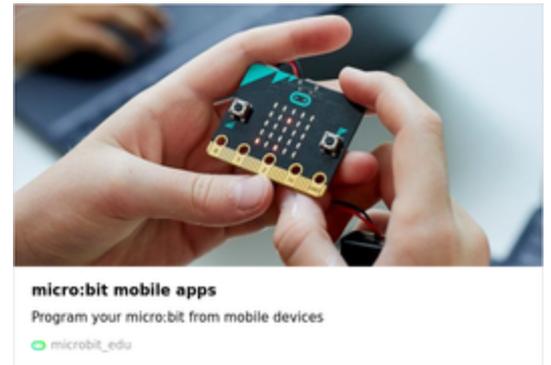
Beide Probleme können normalerweise so gelöst werden:

1. USB-Kabel vom micro:bit abstecken
2. Falls der Akku am micro:bit angesteckt ist: Drücke den Reset-Knopf hinten am micro:bit für 5 Sekunden (oder stecke den Akku kurz ab)
3. USB-Kabel wieder anstecken. Warten bis der Computer den micro:bit erkannt hat und Code noch einmal herunterladen.

Falls das nicht hilft, erstelle ein neues, ganz einfaches, Programm und versuche dieses herunterzuladen.

iPad / Android

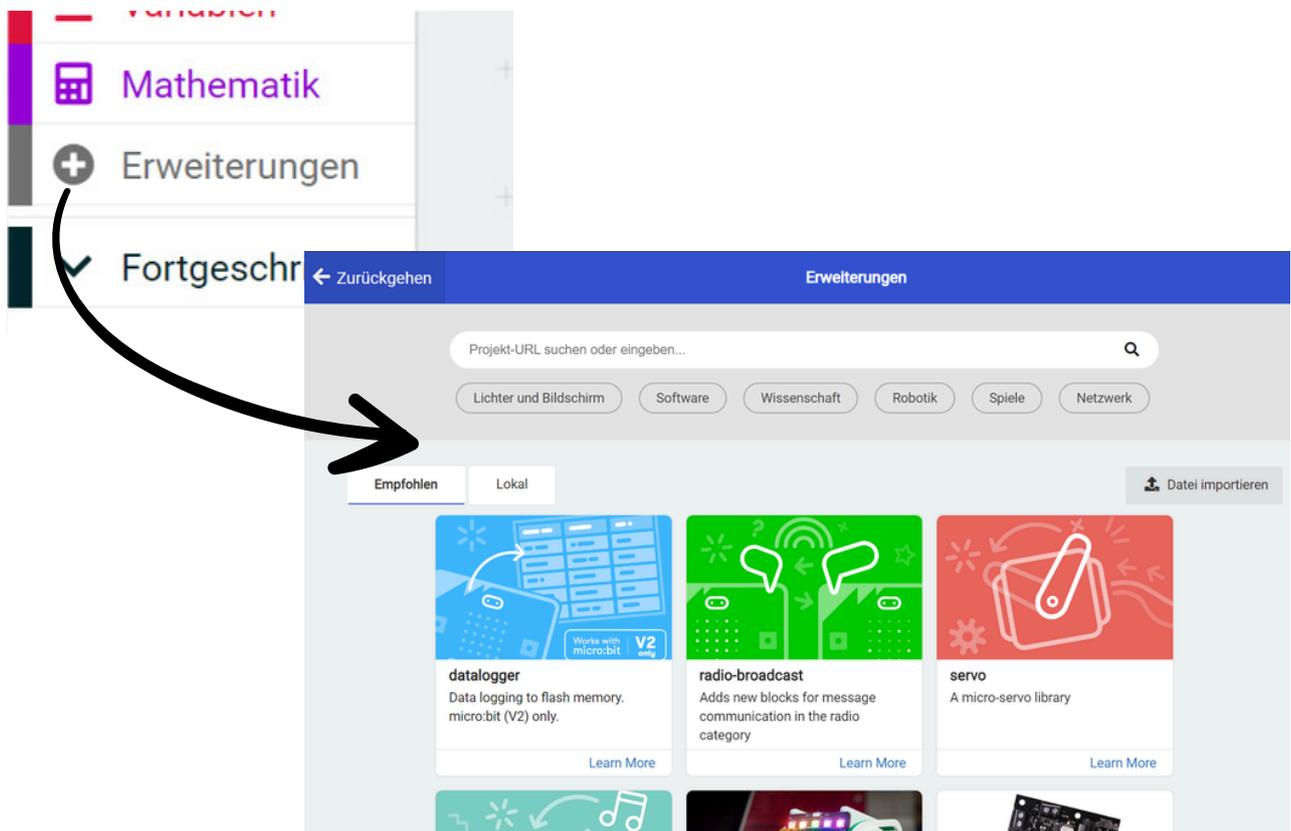
Es ist möglich den micro:bit auch mit einem iPad oder Android Tablet zu programmieren. Das Programm wird dann mit einem Kabel oder Bluetooth auf den micro:bit übertragen. Hierzu sind einige zusätzlich Schritte notwendig, details dazu und Links zu den iPad und Android Apps gibt es hier (leider nur in Englisch):



<https://microbit.org/get-started/user-guide/mobile/>

Erweiterungen / Hardware

Der micro:bit kann mit vielen Hardwareergänzungen erweitert werden. Oft müssen dazu eigene Blöcke geladen werden. Hierzu muss man den Befehl “Erweiterungen” klicken und dann die gewünschte Erweiterung mit Namen suchen und anklicken. Bitte Herstellerinformationen beachten!



LED-Display aus 25 (roten) LEDs

zur Ausgabe von Symbolen, Zahlen, Zeichen oder Text.
Jede LED kann individuell programmiert werden

Lichtsensord

um festzustellen, ob es hell oder dunkel ist

Button A

zur Steuerung des Programmes.
Was soll passieren, wenn der Button gedrückt wird?

Logo als Touch-Sensor

reagiert auf Berührungen und dient, wie die Buttons A und B, der Steuerung des Programmes

Mikrofon

die kleine Öffnung dient der Messung des Lärmpegels

Status-LED für das Mikrofon

zeigt an, ob das Mikrofon aktiv ist

Button B

Ground-Pin

um einen Stromkreis zu schließen

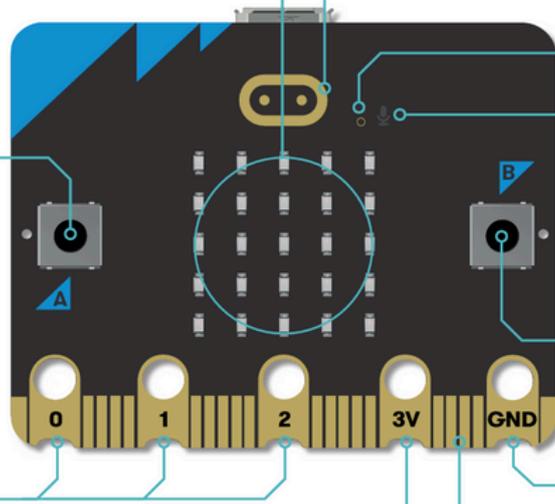
Input/Output-Pins

reagieren auf elektrische Ströme.
Die Pins 0, 1 und 2 können Mikroströme aussenden und empfangen

Power-Pin

hier herrscht ein Spannungspegel von 3,3 Vol

Anschlüsse für weitere elektrische Komponenten
siehe <http://microbit.org/guide/hardware/pins/>



Rückseite

Mikrofon

Temperatursensord

Mikrochip

Prozessor und Speicher, das „Gehirn“ des BBC micro:bit

Kompass

um festzustellen, in welche Richtung der BBC micro:bit zeigt

Beschleunigungssensord

um festzustellen, wie der BBC micro:bit gerade bewegt wird

Lautsprecher

um Töne und Melodien auszugeben

Bluetooth-Antenne

zum Senden und Empfangen von Nachrichten von BBC micro:bit zu BBC micro:bit

Mikro-USB-Anschluss

um den BBC micro:bit mit dem Computer zu verbinden oder mit Strom zu versorgen

Rote Status-LED

leuchtet, wenn die Platine mit Strom versorgt ist

Batterieanschluss

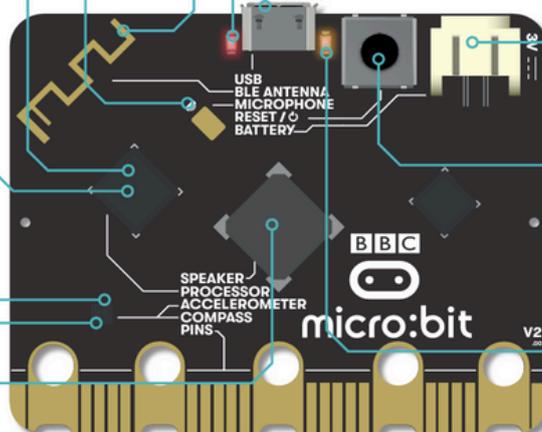
um den BBC micro:bit mit Strom zu versorgen

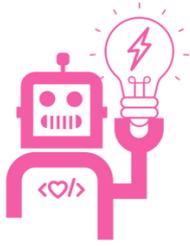
RESET-Button

um ein Programm neu zu starten und in den Stromsparmodus zu wechseln

Gelbe Status-LED

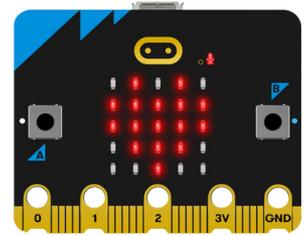
blinkt, wenn ein Programm mittels USB-Kabel auf die Platine übertragen wird





BLÖCKE

AUF DIE FORM KOMMT ES AN!



Wenn Du Dich im MakeCode Editor umschaust dann hast Du vielleicht schon bemerkt dass unsere Blöcke verschiedene Formen haben können. Hier erfährst Du was diese Formen bedeuten:



→ Ich bin ein normaler Befehlsblock. Mich muss man oben an meiner Kerbe irgendwo anhängen!

← An meiner "Nase" kann man noch einen Block anhängen!



→ Ich bin ein Ereignisblock. Ich stehe alleine und kann nicht angehängt werden. In meine Klammer kannst Du viele andere Blöcke stecken. Diese laufen immer dann, wenn mein "Ereignis" eintritt!



→ Ich bin oval und eine Variable. In mir ist ein Wert gespeichert. Den kannst anzeigen oder vergleichen. Ich kann mich verändern - ich bin variabel!

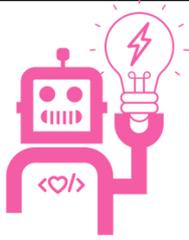


→ Ich bin eine "wahr/falsch" Block. Wenn ich gefragt werde, dann antworte ich nur mit "das ist wahr" oder "das is falsch"! Ist der Knopf 'A' gedrückt? Ich schau nach und sag es Dir!

Graue Blöcke:

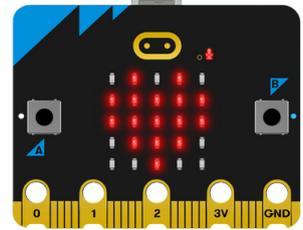


→ Wenn ein Block mit einer Kerbe oben nicht angehängt wird, dann ist er grau und wird nicht benutzt!

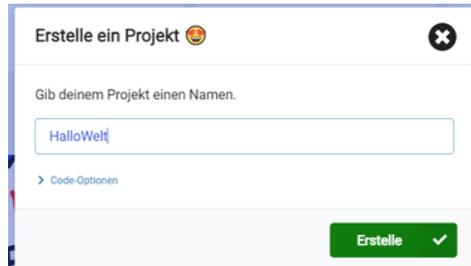


HALLO WELT

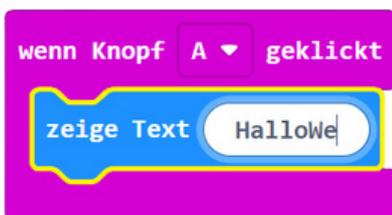
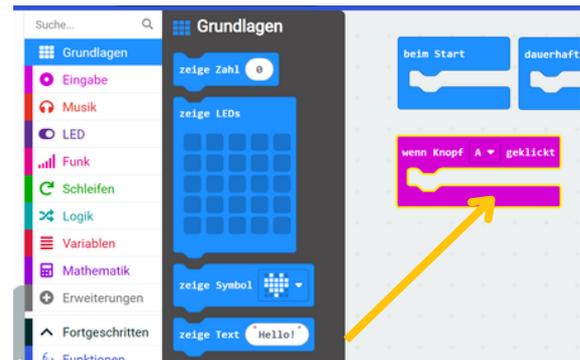
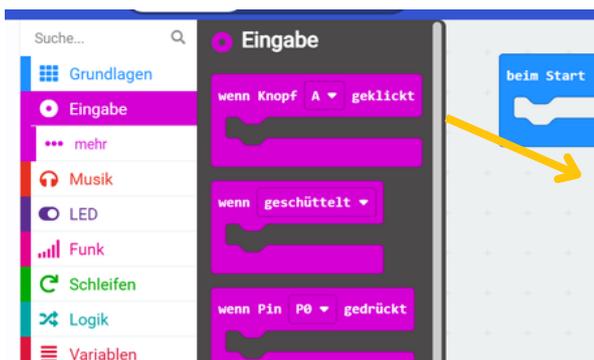
MEIN ERSTES PROGRAMM



- Öffne den MakeCode Editor und klicke auf "Neues Projekt"
- Dann gib dem Projekt einen Namen und klicke "Erstellen"

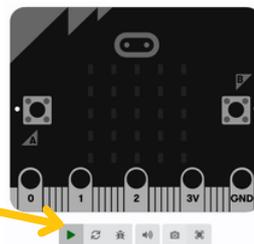


- Ziehe einen "wenn Knopf A geklickt" Block vom "Eingabe" Menü heraus
- Dann vom "Grundlagen" Menü einen "zeige Text" Block. Den klickst Du in den "wenn Knopf A geklickt" Block hinein.



- Ändere den Text auf "Hallo Welt"
- Im micro:bit Simulator kannst Du das testen

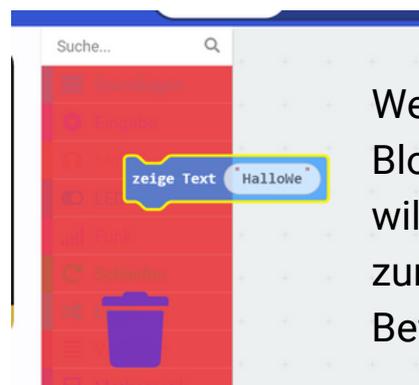
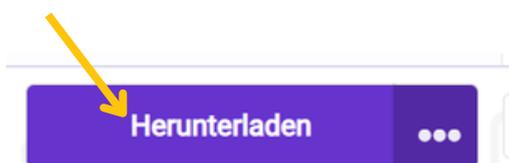
Falls der Simulator noch nicht läuft, klicke den grünen Pfeil.



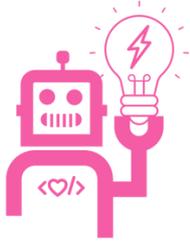
Klicke A um zu testen!



Stecke das USB Kabel an und lade den Code auf deinen micro:bit. Funktioniert es?

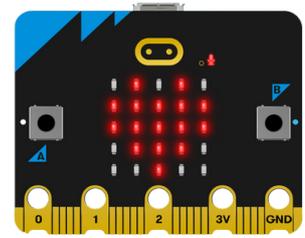


Wenn Du einen Block löschen willst, zieh ihn zurück auf die Befehlsleiste!



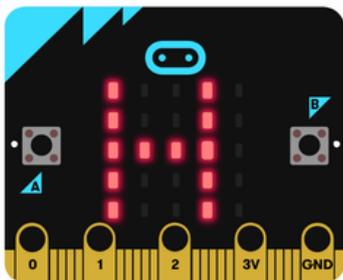
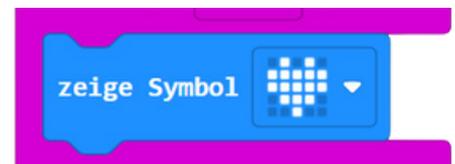
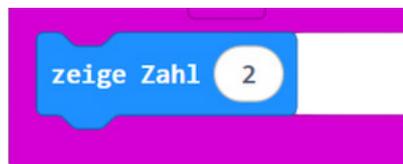
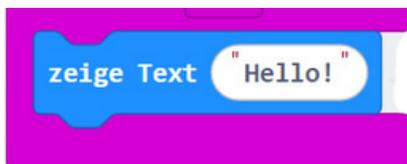
DISPLAY

WIR KÖNNEN ETWAS ANZEIGEN

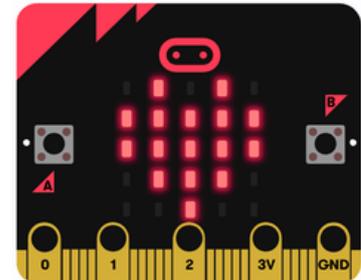
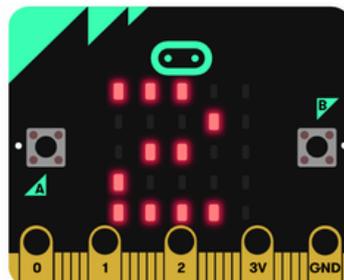


- Der micro:bit hat auf der Vorderseite ein 5x5 LED Licht-Display. Also 25 Lichter in einem Quadrat angeordnet.
- LEDs sind kleine elektronische Bauteile die Strom in farbiges Licht umwandeln können. Je mehr Strom, desto heller!
- Auf diesem Display kann der micro:bit Bilder, Text und Zahlen anzeigen

Hier sind die einfachsten Blöcke dazu:

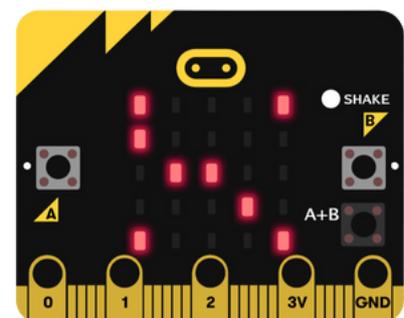


Text scrollt Buchstabe für Buchstabe

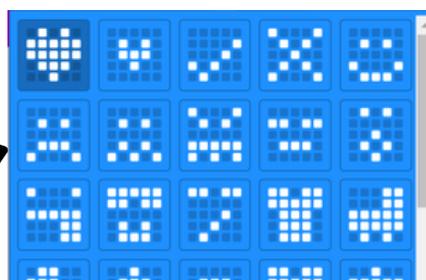
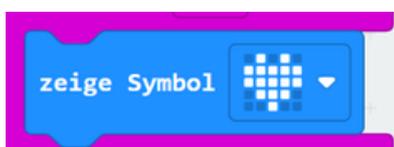


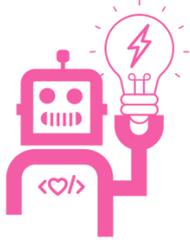
Du kannst auch selbst ein Bild erstellen:

Zum löschen verwende:

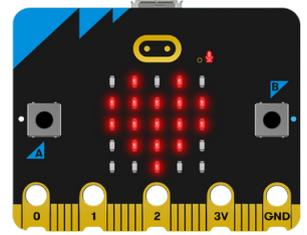


Vorgefertigte Symbole gibt es auch:





TÖNE & MUSIK



JETZT WIRD ES LAUT!

- Grundlagen
- Eingabe
- Musik
- LED

Der micro:bit hat einen Lautsprecher, mit dem können wir Töne abspielen und auch Melodien komponieren! Hier können wir Noten spielen wie auf einem Klavier!



Klicke hier dann siehst Du ein Klavier



Welcher Ton?

Wie lange?

Warten bis zum Ende oder gleich mit dem nächsten Befehl weiter?



Auch Pausen können wir einlegen!

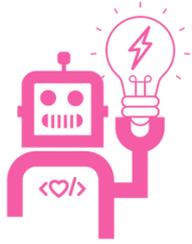
Auch eine Melodie können wir komponieren:



Oder wir spielen eine Melodie die schon komponiert wurde:

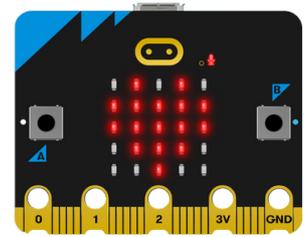


Es gibt noch viele andere Musikblöcke zu entdecken, experimentiere selbst!



EREIGNISSE

SO GEHT'S LOS



- Um überhaupt Code im micro:bit laufen lassen zu können brauchen wir spezielle Blöcke die entscheiden wann unser Code startet!
- Diese Blöcke heißen "Ereignis Blöcke", die "ihren" Code laufen lassen so bald das genannte Ereignis eintritt.
- Ereignis Blöcke stehen allein und können nirgends angeknüpft werden. Sie sind oben flach!
- Sie schauen aus wie eine Klammer. In diese Klammer können wir beliebig viele Blöcke reinklicken.

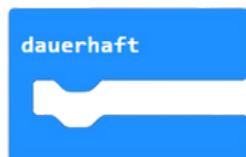


Mein Code läuft wenn der 'A' Knopf gedrückt wird!

Hier ist noch Platz für viele Blöcke!



Läuft **einmal** wenn der micro:bit gestartet (angeschlossen) oder der Reset-Knopf gedrückt wird.



Eine Schleife die immerzu läuft solange der micro:bit eingeschaltet ist.

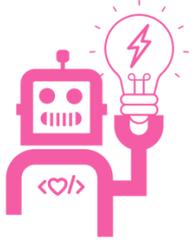


Unser Code läuft, wenn der 'A' Knopf gedrückt wird (oder 'B' oder 'A+B').



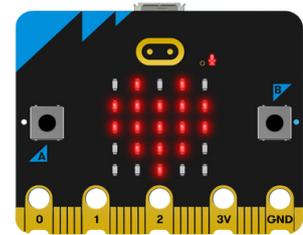
Unser Code läuft wenn der micro:bit geschüttelt wird (oder andere Bewegungen).

Es gibt noch viele andere dieser Blöcke, schau Dich im MakeCode Editor um!



SENSOREN

DAS WEISS ER ALLES



Unser micro:bit ist schlau und kann uns viele Informationen geben. Er hat "Inputs" von seinen Sensoren, die wir abfragen und anzeigen können. Diese Werte werden in ovalen Variablen gespeichert, diese findest Du zum Beispiel unter den "Eingabe" Blöcken:

Lichtstärke

← Wie hell ist es?

Kompassausrichtung (°)

← In welche Richtung schauen wir?

Temperatur (°C)

← Wie heiß / kalt ist es?

Lautstärke

← Wie laut ist es?

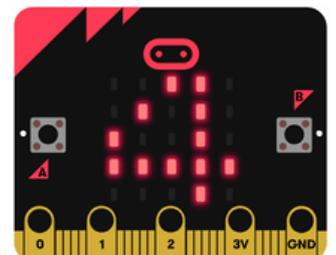
Beschleunigung (mg) x ▾

← Wie schnell werde ich geschüttelt?

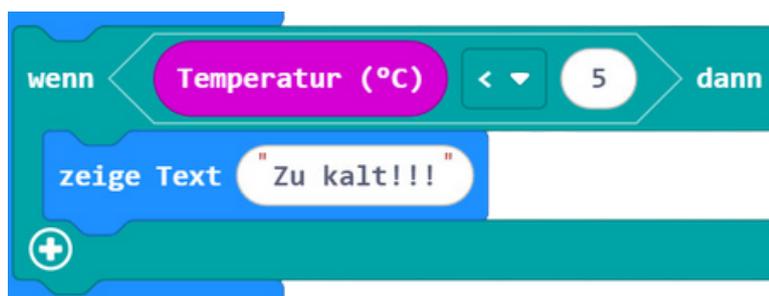
zeige Zahl 0

zeige Zahl Temperatur (°C)

So können wir die Werte anzeigen. Ziehe den ovalen Block in ein ovales Fenster!

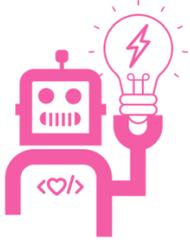


Wir können diese Werte auch vergleichen um Entscheidungen zu treffen:



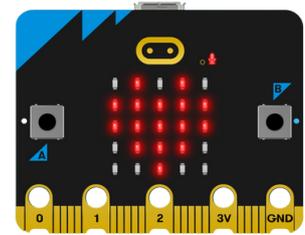
Ziehe die ovalen Temperatur Block in einen Vergleichsblock und vergleiche ihn mit einer Zahl oder einer anderen Variable.

Schau Dir die "Logik" Seite an um mehr zu erfahren!



LOGIK

SCHWERE ENTSCHEIDUNGEN!



Oft müssen wir in unserem Code eine "Logik" einbauen. Manche Befehle sollen nur laufen, wenn etwas anderes passiert ist. Die "Wenn-Dann-Ansonsten! Blöcke helfen uns dabei:



Nur wenn der Bedingungsblock hier "richtig" meldet (also der A Knopf ist gedrückt) dann werden die Blöcke ausgeführt. In diesem Beispiel hören wir einen Ton.

Wenn wir auf das + Symbol drücken dann bekommen wir noch einen Abschnitt. Zuerst den "ansonsten" Teil und dann weitere "wenn-dann" Teile, so viele wie wir brauchen!

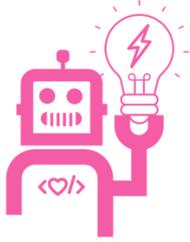


So können wir eine Überprüfung nach der Anderen machen:

- Ist A+B gedrückt?
- Ist A gedrückt?
- Ist B gedrückt?
- Ansonsten (nichts von den Fragen oben trifft zu)

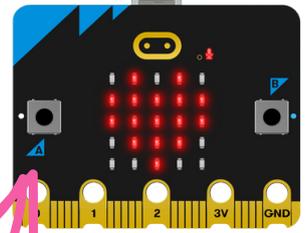
Wenn wir auf das "-" Symbol drücken dann wird diese Überprüfung gelöscht.

WICHTIG: Sobald eine Bedingung also "wahr" gemeldet wird (z.B. "A" Knopf ist gedrückt) dann werden die Befehle in dieser Klammer ausgeführt. Dann geht es aber gleich an das Ende weiter - keine weiteren Bedingungen werden überprüft!!!

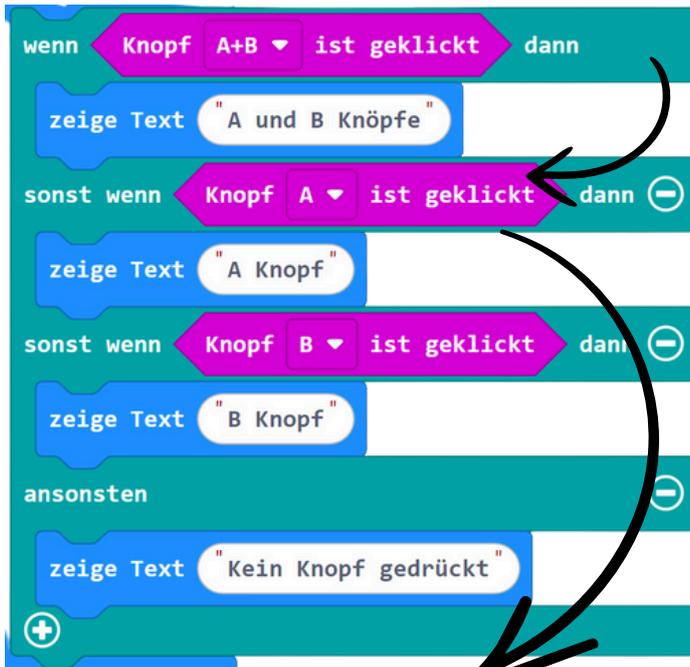


LOGIK

SCHWERE ENTSCHEIDUNGEN!



Beispiel: Der "A" Knopf ist gedrückt



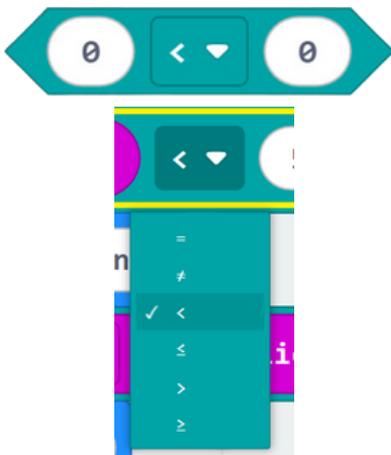
← **Falsch:** nur A gedrückt, es geht mit der nächsten Bedingung weiter

← **Richtig:** A ist gedrückt! Jetzt wird alles übersprungen und es geht ganz unten weiter!

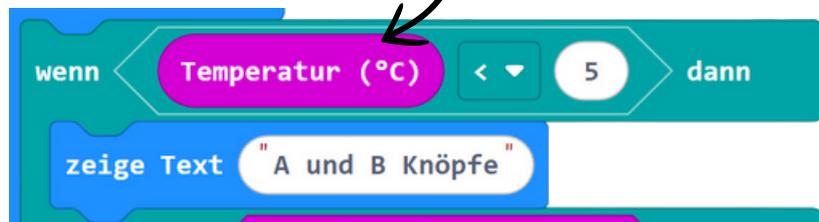
Achtung: Wird nicht geprüft!!!

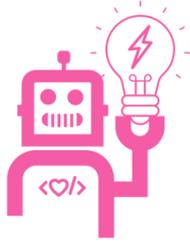
Jetzt werden die nächsten Befehle ausgeführt.

Vergleiche: Oft wollen wir eine Variable vergleichen, um Befehle nur dann auszuführen wenn die Variable einen bestimmten Wert hat. Dazu verwenden wir die spitzen "wahr/falsch" Blöcke:



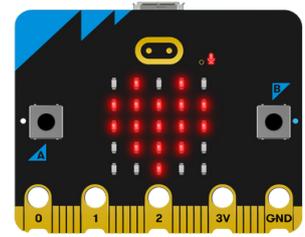
- Hier können wir zwei Werte vergleichen (gleich, größer, kleiner, nicht gleich, etc.)
- Wenn gefragt berichtet dieser Block "ist wahr" oder "ist falsch"
- In das ovale Fenster können wir unsere Variablen reinziehen:





SCHLEIFEN

JETZT GEHT ES RUND!



Oft wollen wir die gleichen Blöcke nicht nur einmal, sondern mehrmals laufen lassen. Hier helfen uns die



Wie oft soll die Schleife sich wiederholen?

x4

4 -mal wiederholen

mache

Note (Hz) Mittleres C

In dieser Schleife können wir entscheiden, wie oft die Blöcke in der Schleife ausgeführt werden, in diesem Beispiel 4-mal.

Unten angekommen geht es wieder zurück zum Anfang, bis die Schleife oft genug gelaufen ist (in diesem Beispiel 4 mal).

Die "Bedingungschleife":

während

Temperatur (°C) < 5

mache

zeige Text "Es ist kalt!"

Bei jedem Durchgang prüfe ich, ob diese Bedingung noch "wahr" ist! Zum Beispiel: Ist es noch kälter als 5 Grad? Wenn ja, dann geht es in der Schleife weiter!

Ich laufe im Kreis solange meine Bedingung ganz oben stimmt, also so lange der "wahr/falsch" Block mir sagt dass es "wahr" ist! Sobald es "falsch" ist, hört die Schleife auf und geht es mit den Blöcken unter mir weiter.

während wahr

mache

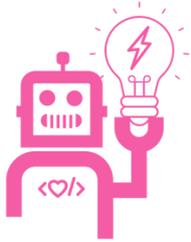
Note (Hz) Mittleres C

wenn Knopf A ist geklickt dann

abbrechen

Bei meiner Bedingung steht einfach "wahr"... Das heißt ich laufe immer weiter, ich bin eine Endlosschleife!

Aus mir kommst Du nur mit einem "abbrechen" Block heraus, mit dem hört die Schleife auf und es geht unten weiter.



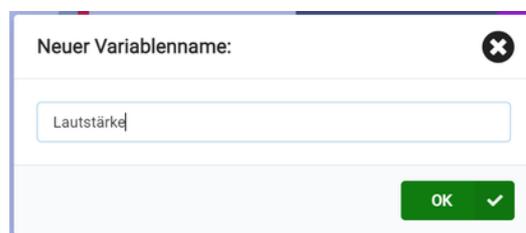
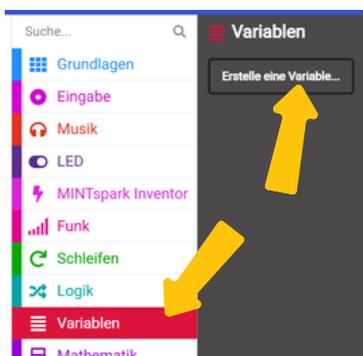
MEINE VARIABLEN

EIGENE VARIABLEN ERSTELLEN

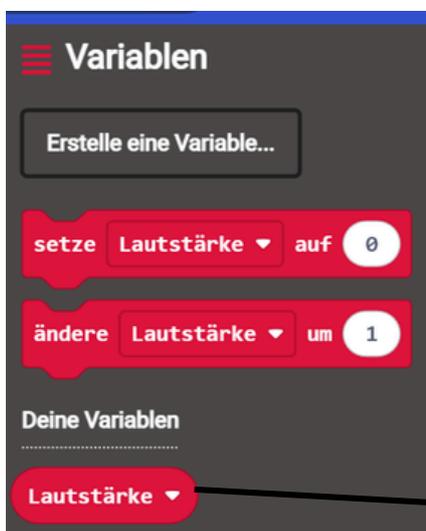
42



- Eine Variable ist ein “Behälter” mit einem Name, der einen Wert (z.B. eine Zahl oder einen Text) speichert
- Während das Programm läuft kann eine Variable geändert werden
- Die Variable kann abgefragt werden: Was ist momentan Dein Wert?
- Wir können unsere eigenen Variablen erstellen:



- Den Namen kannst Du Dir aussuchen
- Er sollte Sinn machen



Setze einen neuen Wert

Verändere den Wert

Benutze den Wert



Zum umbenennen oder löschen Deiner Variable klicke auf den kleinen Pfeil